Trabalho Prático 2
Geração de Números Aleatórios

Grupo 13
Alisson Moreira Ferreira - 11/0106946
Augusto Cesar Ribeiro Nunes - 13/0103004

16 de abril de 2016

Resumo

Este Trabalho Prático implementa um gerador variáveis aleatórias Uniformes no intervalo (0,1) utilizando o Método Congruencial em (3), implementa a geração de variáveis aleatórias Normais padrão utilizando o Método polar em (4.1), e constrói uma tabela para as probabilidades acumuladas da N(0,1) utilizando duas implementações: o Método de Integração de Monte-Carlo em (5) e a Probabilidade Acumulada Empírica obtida em (4) no item (6). Os resultados são discutidos em 7.

1 Introdução

Técnicas de Simulação são amplamente utilizadas em Ciências Naturais, Biológicas e na Tecnologia. Os seus custos diminuíram drasticamente nas últimas décadas, e acompanharam a evolução da computação em sentido inverso: hoje podemos simular processos com alto grau de sensibilidade e especificidade até mesmo em nossos computadores pessoais, e não mais em *mainframes* cujo acesso era restrito e o custo elevadíssimo.

Nas Ciências Físicas, podemos citar os sofisticados Modelos Climáticos (RAMÍREZ; ORSINI, 2007) que utilizam simulação para a obtenção de previsões com maior ou menor grau de sucesso[1]. Na Estatística, a aplicação mais conhecida de Técnicas de Simulação são as ligadas direta ou indiretamente aos chamados **Métodos de Monte-Carlo**.

Na Estatística, encontramos aplicações (sofisticadas) de Simulação em Técnicas de Redução de Variância (como Amostragem de Importância e Amostragem Antitética) e em Processos Estocásticos (Algoritmo de Metropolis-Hastings, Amostragem de Gibbs, e MCMC). Em um nível mais simples, qualquer programa de computação científica que se preze contém uma implementação de geradores de variáveis aleatórias, um uso mais simples de Simulação.

2 Metodologia

Este Trabalho Prático aborda em um de seus tópicos a Integração Numérica utilizando Monte-Carlo. A ideia é a seguinte: sendo $g(x)$ uma função arbitrária, e supondo que desejamos obter uma aproximação numérica para θ onde

$$\theta = \int_0^1 g(x)dx \tag{1}$$

podemos olhar para θ como a Esperança de uma função $g(U)$ com $U \sim U(0,1)$. Supondo $U_1, \ldots, U_n$ variáveis aleatórias independentes e identicamente distribuídas com $U_i \sim U(0,1)$ para $i = 1, \ldots, n$, segue que $g(U_1), \ldots, g(U_n)$ são variáveis aleatórias independentes e identicamente distribuídas com média θ. Portanto, pela Lei Forte dos Grandes Números, temos com Probabilidade 1 que

$$\lim_{n\to\infty} \sum_{i=1}^{n} \frac{g(U_i)}{n} \to E[g(U)] = \theta, \tag{2}$$

Logo, nós podemos aproximar θ utilizando uma amostra *grande* de u_i tirados de uma distribuição Uniforme em $(0,1)$. Para o caso mais geral onde os limites inferior e superior da integral (1) são a e b quaisquer $(a < b)$, podemos aplicar a substituição $y = (x-a)/(b-a), dy = dx/(b-a)$, obtendo

$$\theta = \int_0^1 g(a + [b-a]y)(b-a)dy = \int_0^1 h(y)dy \tag{3}$$

[1] O próprio *paper* em questão faz uma espécie de meta-análise de modelos do IPCC, GFDL e HAD

onde $h(y) = (b-a)g(a+[b-a]y)$. Portanto, podemos aproximar θ tomando uma amostra de variáveis aleatórias Uniformes em $(0,1)$ e tomando a média de h avaliada nas observações da amostra. A última parte do trabalho consiste na criação de uma tabela para a densidade acumulada da Normal(0,1), i.e., vamos aproximar a integral $\int_0^b \frac{1}{2\pi} exp\{-\frac{x_i}{2\sigma^2}\}$ utilizando o Método de Monte Carlo para integração numérica. O método foi implementado pela função **pnorm_MC()**, e os valores obtidos com este método são comparados com os obtidos pela função **pnorm()** do R.

Inicialmente há a necessidade de implementação de um gerador de variáveis aleatórias para a distribuição Uniforme em $(0,1)$. Utilizou-se o **Método Congruencial Multiplicativo** para tal finalidade. O método em questão consiste em, a partir de uma semente *aleatória*[2], obter uma sequência de números aleatórios $x_1, x_2, \ldots$ tomando $x_n = (ax_{n-1})$ mod m, com a e m inteiros positivos. Este gerador foi implementado pela função **runif_congruencial()**.

Um outro método para geração de variáveis aleatórias estudado neste Trabalho Prático é o Método Polar[3]: este método gera pares de variáveis aleatórias contínuas utilizando uma transformação em coordenadas polares. Em particular, implementamos um gerador de variáveis aleatórias Normais padrão, e sua função é **rnorm_polar()**. Em um segundo momento, a distribuição acumulada empírica utilizando este método também foi utilizada para gerar uma tabela dos quantis da Normal padrão, comparando-a com os resultados da função **pnorm()** do R.

As implementações, os resultados dos testes de hipóteses e sementes utilizados para verificar o ajustamento dos geradores encontram-se no Anexo A. As funções foram documentadas utilizando o pacote *Roxygen* do R. Disponibilizamos um repositório no site github.com: `http://github.com/august-o/TP2-EstComp`. Lá encontram-se todos os arquivos-fonte utilizados neste Trabalho Prático. O computador utilizado foi um Macbook Pro, Sistema Operacional OS X El Capitan (10.11.3), com processador Intel Core i5 de 2.4 GHz e 16 GB de memória RAM 1333 MHz DDR3. A versão do R utilizada foi a 3.2.4, com a IDE RStudio em sua versão 0.99.1130.

[2] Na verdade, a maior parte os geradores aleatórios implementados e utilizados habitualmente são pseudo-aleatórios. Ver (WIKIPEDIA, 2015) [em inglês] para uma explicação mais detalhada.

[3] Em particular, utilizamos aqui o que se chama de Método Polar de Marsaglia: ver

3 Gerador Congruencial para a variável aleatória Uniforme(0,1)

3.1 Teste de Uniformidade

Com a implementação adotada e a amostra obtida (ver Anexo A), foram realizados três Testes de Hipóteses para

$$H_0 : \mathbf{X} \sim U(0,1), (iid)$$
$$H_1 : \mathbf{X} \sim T$$

onde $\mathbf{X}$ é o vetor aleatório ($n = 10^5$) gerado pela implementação utilizada, e T é uma distribuição qualquer que não seja a Uniforme(0,1). O **Teste de Kolmogorov-Smirnov** para a Uniformidade da Distribuição gerada não rejeita a hipótese nula a um nível de significância de 0.7253, o **Teste de Cramér-von Mises** não rejeita a hipótese nula a um nível de significância 0.7823, e o **Teste de Anderson-Darling** não rejeita a hipótese nula a um nível de significância de 0.5601.

Como verificação qualitativa, adicional, abaixo estão os histogramas do vetor usando o gerador congruencial e do vetor utilizando **runif()** do R. Note que ambos são *uniformes* e muito similares[4]

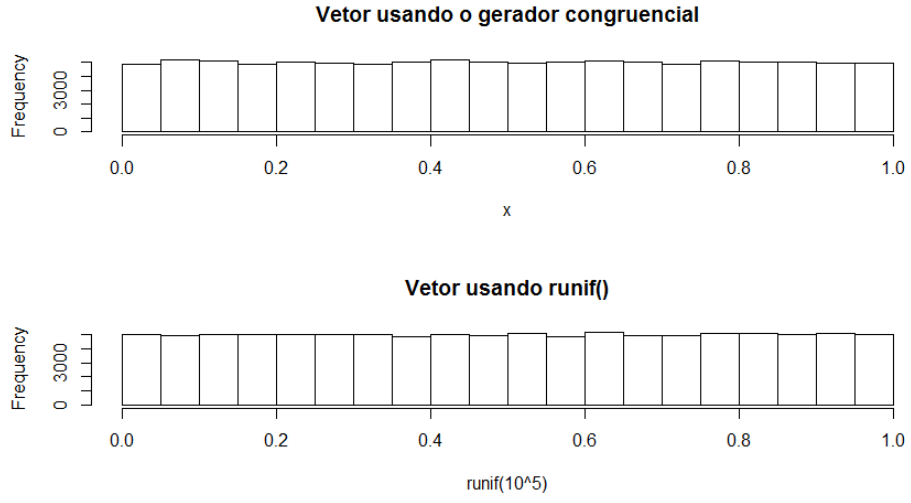

Figura 1 – Histogramas para o gerador e para a função runif(), $n = 10^5$

[4] Isso é o máximo que se pode dizer numa inspeção visual para "verificar"esse tipo de similaridade.

4 Gerador Polar para a variável aleatória Normal Padrão

4.1 Teste de Normalidade

Com a implementação adotada e a amostra obtida (ver Anexo A), foram realizados três Testes de Hipóteses para

$$H_0 : \mathbf{X} \sim N(0,1), (iid)$$
$$H_1 : \mathbf{X} \sim T$$

onde $\mathbf{X}$ é o vetor aleatório ($n = 10^7$) gerado pela implementação utilizada, e T é uma distribuição qualquer que não seja a Normal(0,1) iid. O **Teste de Kolmogorov-Smirnov** para a Uniformidade da Distribuição gerada não rejeita a hipótese nula a um nível de significância de 0.8173, o **Teste de Cramér-von Mises** não rejeita a hipótese nula a um nível de significância 0.8573, e o **Teste de Anderson-Darling** não rejeita a hipótese nula a um nível de significância de 0.8223.

Como verificação qualitativa, adicional, abaixo estão os gráficos de densidade *Kernel* do vetor usando o Método Polar de Marsaglia e do vetor utilizando **rnorm()** do R.

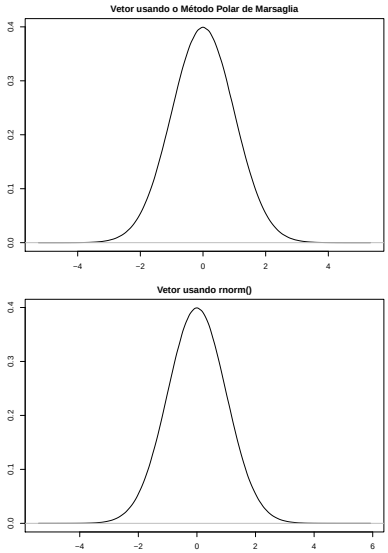

Figura 2 – Gráficos de Densidade *Kernel* para o gerador que usa o Método Polar de Marsaglia e a para a função rnorm(), $n = 10^7$ em ambos

5 Tabela de densidades acumuladas da Normal(0,1) usando Método de Integração de Monte Carlo

As tabelas a seguir fornecem valores para os percentis (0,b) da Normal Padrão, ou seja, cada célula contém o valor de P(0 < Z < b) com $Z \sim Normal(0, 1)$, e entre parênteses, o valor encontrado via MC menos o valor obtido pela pnorm() do R para o respectivo quantil.

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	0.500000 (0.00e+00)	0.503989 (-4.95e-11)	0.507978 (-3.96e-10)	0.511966 (-1.34e-09)	0.515953 (-3.17e-09)	0.519939 (6.18e-09)	0.523922 (-1.07e-08)	0.527903 (-1.70e-08)	0.531881 (-2.53e-08)	0.535856 (-3.60e-08)
0.1	0.539828 (-7.13e-08)	0.543795 (-9.49e-08)	0.547758 (-1.23e-07)	0.551717 (-1.56e-07)	0.555670 (-1.95e-07)	0.559618 (-2.40e-07)	0.563559 (-2.91e-07)	0.567495 (-3.49e-07)	0.561424 (-4.14e-07)	0.575345 (-4.87e-07)
0.2	0.579260 (-3.42e-07)	0.583166 (-3.95e-07)	0.587064 (-4.54e-07)	0.590954 (-5.19e-07)	0.594835 (5.89e-07)	0.598706 (-6.65e-07)	0.602568 (-7.47e-07)	0.602568 (-8.36e-07)	0.610261 (-9.31e-07)	0.614092 (-1.03e-06)
0.3	0.617912 (1.14e-06)	0.621720 (1.26e-06)	0.625517 (1.38e-06)	0.629301 (1.52e-06)	0.633073 (1.66e-06)	0.636832 (1.81e-06)	0.640578 (1.96e-06)	0.644310 (2.13e-06)	0.648029 (2.30e-06)	0.651733 (2.49e-06)
0.4	0.655427 (-4.64e-07)	0.659103 (-4.99e-07)	0.662763 (-5.36e-07)	0.666409 (-5.75e-07)	0.670038 (-6.15e-07)	0.673652 (-6.57e-07)	0.677250 (-7.01e-07)	0.680831 (-7.47e-07)	0.684395 (-7.95e-07)	0.687942 (-8.45e-07)
0.5	0.691453 (-2.13e-06)	0.694965 (-2.27e-06)	0.698458 (-2.41e-06)	0.701933 (-2.56e-06)	0.705390 (-2.71e-06)	0.708828 (-2.87e-06)	0.712248 (-3.04e-06)	0.715648 (-3.21e-06)	0.719029 (-3.39e-06)	0.722390 (-3.58e-06)
0.6	0.725735 (-1.33e-05)	0.729057 (-1.39e-05)	0.732358 (-1.46e-05)	0.735639 (-1.53e-05)	0.738900 (-1.60e-05)	0.742139 (-1.67e-05)	0.745358 (-1.74e-05)	0.748555 (-1.82e-05)	0.751731 (-1.90e-05)	0.754886 (-1.97e-05)
0.7	0.758042 (-3.35e-05)	0.761153 (-3.48e-05)	0.764243 (-3.62e-05)	0.767311 (-3.76e-05)	0.770356 (-3.90e-05)	0.773379 (-4.05e-05)	0.776379 (-4.19e-05)	0.779357 (-4.34e-05)	0.782312 (-4.50e-05)	0.785244 (-4.65e-05)
0.8	0.788137 (5.09e-05)	0.791022 (5.26e-05)	0.793884 (5.45e-05)	0.796722 (5.63e-05)	0.799537 (5.82e-05)	0.802329 (6.01e-05)	0.805096 (6.20e-05)	0.807840 (6.40e-05)	0.810561 (6.60e-05)	0.813257 (6.81e-05)
0.9	0.815915 (-1.78e-05)	0.818563 (-1.84e-05)	0.821187 (-1.90e-05)	0.823787 (-1.95e-05)	0.826363 (-2.01e-05)	0.828915 (-2.07e-05)	0.831442 (-2.13e-05)	0.833946" (-2.19e-05)	0.836425 (-2.26e-05)	0.838880 (-2.32e-05)
1.0	0.841236 (6.85e-05)	0.843641 (7.03e-05)	0.846022 (7.21e-05)	0.848378 (7.40e-05)	0.850710 (7.59e-05)	0.853018 (7.78e-05)	0.855302 (7.97e-05)	0.857562 (8.17e-05)	0.859797 (8.37e-05)	0.862009 (8.57e-05)
1.1	0.864367 (6.11e-05)	0.866534 (6.25e-05)	0.868677 (6.38e-05)	0.870797 (6.52e-05)	0.872892 (6.66e-05)	0.874964 (6.80e-05)	0.877012 (6.94e-05)	0.879037 (7.08e-05)	0.881038 (7.22e-05)	0.883016 (7.37e-05)
1.2	0.884832 (-9.83e-05)	0.88676 (0.00183)	0.888665 (0.003735)	0.890547 (0.005616)	0.892405 (0.007475)	0.894241 (0.009311)	0.896054 (0.011123)	0.897844 (0.012913)	0.899611 (0.014681)	0.901356 (0.016426)
1.3	0.903219 (1.99e-05)	0.904922 (1.72e-03)	0.906603 (3.40e-03)	0.908262 (5.06e-03)	0.909898 (6.70e-03)	0.911513 (8.31e-03)	0.913107 (9.91e-03)	0.914679 (0.011479)	0.916229 (1.30e-02)	0.917758 (1.46e-02)
1.4	0.919273 (3.01e-05)	0.920761 (1.52e-03)	0.922227 (2.98e-03)	0.923673 (4.43e-03)	0.925098 (0.005855)	0.926503 (7.26e-03)	0.927888 (0.008645)	0.929253 (1.00e-02)	0.930597 (1.14e-02)	0.931922 (1.27e-02)
1.5	0.933274 (0.000081)	0.934561 (1.37e-03)	0.935828 (2.64e-03)	0.937077 (3.88e-03)	0.938306 (5.11e-03)	0.939517 (6.32e-03)	0.940709 (7.52e-03)	0.941883 (8.69e-03)	0.943039 (9.85e-03)	0.944176 (1.10e-02)
1.6	0.945156 (-4.47e-05)	0.946256 (1.05e-03)	0.947338 (2.14e-03)	0.948402 (3.20e-03)	0.949449 (0.004249)	0.95048 (5.28e-03)	0.951493 (6.29e-03)	0.95249 (7.29e-03)	0.95347 (8.27e-03)	0.954434 (9.23e-03)

Tabela 1 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Método de Integração de Monte Carlo. Nos parênteses, o valor encontrado via MC menos o valor obtido pela pnorm() do R para o respectivo quantil

1.7	0.95544 (5.27e-06)	0.956372 (9.38e-04)	0.957289 (1.85e-03)	0.95819 (2.76e-03)	0.959075 (3.64e-03)	0.959946 (4.51e-03)	0.960801 (5.37e-03)	0.961641 (6.21e-03)	0.962467 (7.03e-03)	0.963278 (7.84e-03)
1.8	0.963982 (-8.72e-05)	0.964764 (0.000694)	0.965532 (1.46e-03)	0.966285 (2.22e-03)	0.967025 (2.96e-03)	0.967752 (3.68e-03)	0.968465 (4.40e-03)	0.969165 (5.10e-03)	0.969852 (5.78e-03)	0.970526 (6.46e-03)
1.9	0.971561 (0.000278)	0.972214 (0.000930)	0.972854 (0.001571)	0.973482 (0.002199)	0.974098 (0.002815)	0.974702 (0.003419)	0.975295 (0.004012)	0.975876 (0.004593)	0.976446 (0.005163)	0.977005 (0.005721)
2.0	0.977237 (-1.28e-05)	0.977771 (5.22e-04)	0.978295 (1.05e-03)	0.978809 (1.56e-03)	0.979311 (2.06e-03)	0.979804 (2.55e-03)	0.980287 (3.04e-03)	0.98076 (3.51e-03)	0.981223 (3.97e-03)	0.981677 (4.43e-03)
2.1	0.982103 (-3.28e-05)	0.982537 (4.02e-04)	0.982963 (8.28e-04)	0.98338 (1.24e-03)	0.983788 (1.65e-03)	0.984187 (2.05e-03)	0.984578 (2.44e-03)	0.98496 (2.82e-03)	0.985334 (0.003199)	0.9857 (3.56e-03)
2.2	0.986247 (0.000151)	0.986601 (0.000504)	0.986946 (0.000850)	0.987284 (0.001188)	0.987615 (0.001518)	0.987938 (0.001841)	0.988254 (0.002158)	0.988563 (0.002467)	0.988866 (0.002769)	0.989161 (0.003065)
2.3	0.989135 (-0.00014)	0.989414 (0.000138)	0.989687 (0.000411)	0.989954 (0.000678)	0.990214 (0.000938)	0.990469 (0.001193)	0.990718 (0.001442)	0.99096 (1.68e-03)	0.991198 (1.92e-03)	0.991429 (2.15e-03)
2.4	0.991659 (-0.00014)	0.991879 (0.000077)	0.992094 (2.92e-04)	0.992304 (5.02e-04)	0.992509 (7.06e-04)	0.992709 (9.06e-04)	0.992903 (1.10e-03)	0.993094 (1.29e-03)	0.993279 (1.48e-03)	0.99346 (0.001657)
2.5	0.993762 (-2.87e-05)	0.993936 (1.46e-04)	0.994106 (3.15e-04)	0.994272 (4.81e-04)	0.994433 (6.43e-04)	0.994591 (0.000801)	0.994745 (9.54e-04)	0.994894 (1.10e-03)	0.995041 (1.25e-03)	0.995183 (1.39e-03)
2.6	0.994795 (-0.000544)	0.994927 (-0.000412)	0.995056 (-0.000283)	0.995181 (-0.000158)	0.995303 (-3.59e-05)	0.995422 (8.28e-05)	0.995537 (1.98e-04)	0.99565 (3.11e-04)	0.995759 (4.20e-04)	0.995866 (5.27e-04)
2.7	0.997312 (7.79e-04)	0.997419 (8.86e-04)	0.997524 (0.000991)	0.997625 (0.001092)	0.997724 (0.001191)	0.997821 (0.001288)	0.997914 (0.001381)	0.998006 (0.001473)	0.998095 (0.001562)	0.998182 (0.001649)
2.8	0.997993 (5.48e-04)	0.998072 (6.27e-04)	0.998148 (7.03e-04)	0.998223 (0.000778)	0.998295 (0.000850)	0.998366 (0.000921)	0.998434 (0.000989)	0.998501 (0.001056)	0.998565 (0.001120)	0.998628 (0.001183)
2.9	0.997888 (-0.000246)	0.997945 (-0.000190)	0.997999 (-0.000135)	0.998053 (-8.16e-05)	0.998104 (-3.00e-05)	0.998154 (2.00e-05)	0.998203 (6.84e-05)	0.99825 (1.15e-04)	0.998295 (1.61e-04)	0.998339 (2.05e-04)
3.0	0.99867 (2.03e-05)	0.998713 (6.28e-05)	0.998754 (1.04e-04)	0.998794 (1.44e-04)	0.998833 (1.83e-04)	0.99887 (2.20e-04)	0.998907 (2.56e-04)	0.998942 (2.92e-04)	0.998976 (3.26e-04)	0.999009 (3.59e-04)
3.1	1.00007 (0.001039)	1.00011 (0.001074)	1.00014 (0.001107)	1.00017 (0.001140)	1.0002 (0.001172)	1.00023 (0.001202)	1.00026 (0.001232)	1.00029 (0.00126)	1.00032 (0.00129)	1.00035 (0.00132)
3.2	0.999571 (2.58e-04)	0.999596 (2.83e-04)	0.99962 (0.000307)	0.999644 (0.000331)	0.999667 (0.000354)	0.999689 (0.000376)	0.99971 (0.000397)	0.999731 (0.000418)	0.999751 (0.000438)	0.99977 (0.000458)

Tabela 2 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Método de Integração de Monte Carlo. Nos parênteses, o valor encontrado via MC menos o valor obtido pela pnorm() do R para o respectivo quantil (continuação)

3.3	0.998938 (-0.000579)	0.998952 (-0.000564)	0.998966 (-0.000551)	0.998979 (-0.000538)	0.998992 (-0.000525)	0.999004 (-0.000513)	0.999016 (-0.000501)	0.999027 (-0.000490)	0.999037 (-0.000479)	0.999048 (-0.000469)
3.4	0.999377 (-0.000286)	0.999389 (-0.000274)	0.999399 (-0.000264)	0.99941 (-0.000253)	0.99942 (-0.000243)	0.99943 (-0.000233)	0.999439 (-0.000224)	0.999448 (-0.000215)	0.999457 (-0.000207)	0.999465 (-0.000198)
3.5	0.999125 (-0.000642)	0.999133 (-0.000635)	0.99914 (-0.000628)	0.999147 (-0.000621)	0.999153 (-0.000614)	0.999159 (-0.000608)	0.999165 (-0.000602)	0.999171 (-0.000596)	0.999177 (-0.000591)	0.999182 (-0.000586)
3.6	1.00103 (0.00119)	1.00104 (0.00120)	1.00105 (0.00121)	1.00106 (0.00122)	1.00107 (0.00123)	1.00108 (0.00124)	1.00109 (0.00125)	1.0011 (0.00126)	1.00111 (0.00127)	1.00112 (0.00127)
3.7	1.00019 (0.000297)	1.00019 (0.000301)	1.0002 (0.000305)	1.0002 (0.000310)	1.00021 (0.000313)	1.00021 (0.000317)	1.00021 (0.000321)	1.00022 (0.000325)	1.00022 (0.000328)	1.00022 (0.000331)
3.8	0.999315 (-0.000613)	0.999316 (-0.000612)	0.999317 (-0.000611)	0.999318 (-0.000610)	0.999319 (-0.000609)	0.999319 (-0.000608)	0.99932 (-0.000608)	0.99932 (-0.000607)	0.999321 (-0.000607)	0.999321 (-0.000606)
3.9	0.999732 (-0.000220)	0.999733 (-0.000219)	0.999735 (-0.000217)	0.999736 (-0.000216)	0.999737 (-0.000215)	0.999738 (-0.000214)	0.999739 (-0.000213)	0.99974 (-0.000212)	0.999741 (-0.000211)	0.999742 (-0.000210)

Tabela 3 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Método de Integração de Monte Carlo. Nos parênteses, o valor encontrado via MC menos o valor obtido pela pnorm() do R para o respectivo quantil (continuação)

6 Tabela de densidades acumuladas da Normal(0,1) usando a Densidade Acumulada Empírica da Transformação Polar de Marsaglia

As tabelas a seguir fornecem valores para os percentis (0,b) da Normal Padrão, ou seja, cada célula contém o valor de P(0 < Z < b) com $Z \sim Normal(0, 1)$, e entre parênteses, o valor encontrado via Transformação Polar menos o valor obtido pela pnorm() do R para o respectivo quantil.

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	0.50211 (0.00211)	0.5059 (0.00590)	0.509915 (0.00992)	0.51402 (0.01402)	0.518225 (0.01823)	0.52202 (0.02202)	0.52595 (0.02595)	0.5298 (0.02980)	0.533455 (0.03346)	0.53741 (0.03741)
0.1	0.539615 (-0.000213)	0.543885 (4.06e-03)	0.547805 (7.98e-03)	0.551995 (0.012167)	0.556 (0.01617)	0.559965 (0.020137)	0.563855 (0.024027)	0.5679 (0.028072)	0.57177 (0.031942)	0.575645 (0.03582)
0.2	0.57933 (7.03e-05)	0.58321 (3.95e-03)	0.587185 (0.007925)	0.59092 (1.17e-02)	0.594985 (0.01573)	0.599035 (0.019775)	0.60298 (0.023720)	0.606835 (0.027575)	0.6109 (0.031640)	0.61476 (0.035500)
0.3	0.6168 (-0.00111)	0.620305 (0.00239)	0.62406 (0.00615)	0.62796 (0.01005)	0.63181 (0.01390)	0.6356 (0.01769)	0.6394 (0.02149)	0.64314 (0.02523)	0.64702 (0.02911)	0.65086 (0.032949)
0.4	0.65609 (0.000668)	0.659765 (0.004343)	0.66359 (0.008168)	0.667305 (0.011883)	0.67067 (0.015248)	0.67425 (0.018828)	0.67782 (0.022398)	0.681515 (0.026093)	0.68494 (0.029518)	0.68843 (0.033008)
0.5	0.691965 (0.000503)	0.69548 (0.004018)	0.69895 (0.007488)	0.70226 (0.010798)	0.70561 (0.014148)	0.70899 (0.01753)	0.71255 (0.02109)	0.716215 (0.024753)	0.71964 (0.028178)	0.722945 (0.03148)
0.6	0.72452 (-0.00123)	0.72782 (0.00207)	0.731275 (0.00553)	0.734715 (0.008968)	0.73792 (0.012173)	0.74133 (0.015583)	0.74458 (0.018833)	0.7478 (0.022053)	0.75095 (0.025203)	0.754075 (0.028328)
0.7	0.757555 (-0.000481)	0.760655 (0.002619)	0.76368 (0.005644)	0.76657 (0.008534)	0.76955 (0.01151)	0.77277 (0.014734)	0.775645 (0.017609)	0.778625 (0.020589)	0.781465 (0.02343)	0.78418 (0.02614)
0.8	0.78875 (0.000605)	0.79151 (0.00337)	0.79447 (0.006325)	0.797215 (0.009070)	0.799785 (0.011640)	0.80253 (0.014385)	0.805245 (0.01710)	0.80788 (1.97e-02)	0.810395 (0.022250)	0.81328 (2.51e-02)
0.9	0.81577 (-0.00017)	0.818465 (0.002525)	0.82102 (0.005080)	0.823585 (0.007645)	0.82631 (1.04e-02)	0.828825 (0.012885)	0.83143 (1.55e-02)	0.834105 (0.018165)	0.83657 (0.020630)	0.83902 (0.023080)
1.0	0.841835 (0.00049)	0.84418 (0.002835)	0.846525 (0.005180)	0.848925 (0.00758)	0.85132 (0.00998)	0.85382 (0.012475)	0.85613 (0.014785)	0.85844 (0.01710)	0.860725 (0.019380)	0.862755 (0.021410)
1.1	0.865295 (0.000961)	0.86741 (0.00308)	0.869395 (0.005061)	0.871665 (0.007331)	0.8738 (0.009466)	"0.87585 (0.011516)	0.877935 (0.013601)	0.87975 (0.01542)	0.88159 (0.01726)	0.8836 (0.019266)
1.2	0.884855 (-7.53e-05)	0.886775 (1.84e-03)	0.88869 (3.76e-03)	0.89043 (0.005500)	0.892165 (0.007235)	0.89397 (0.00904)	0.89586 (0.010930)	0.897665 (0.012735)	0.89931 (0.014380)	0.90109 (0.016160)
1.3	0.90415 (0.000950)	0.905985 (0.002785)	0.90763 (0.004430)	0.9093 (0.006100)	0.91108 (0.007880)	0.91249 (0.009290)	0.91414 (0.010940)	0.915575 (0.012375)	0.917055 (0.013855)	0.91877 (0.01557)
1.4	0.919235 (-8.34e-06)	0.92072 (1.48e-03)	0.922175 (2.93e-03)	0.923545 (4.30e-03)	0.925075 (5.83e-03)	0.92648 (7.24e-03)	0.928045 (0.00880)	0.92946 (0.010217)	0.930965 (0.011722)	0.93231 (0.013067)
1.5	0.93331 (0.000117)	0.93457 (1.38e-03)	0.93585 (0.002657)	0.937165 (0.003972)	0.938455 (0.005262)	0.93956 (0.006367)	0.9406 (7.41e-03)	0.941745 (8.55e-03)	0.94284 (0.009647)	0.944 (1.08e-02)
1.6	0.94567 (0.000469)	0.94674 (0.001539)	0.947855 (0.002654)	0.948955 (0.003754)	0.950085 (0.004884)	0.95115 (0.005949)	0.952155 (0.006954)	0.953145 (0.007944)	0.95417 (0.008969)	0.95523 (0.010029)

Tabela 4 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Transformação Polar de Marsaglia. Nos parênteses, o valor encontrado via Transformação Polar menos o valor obtido pela pnorm() do R para o respectivo quantil (continuação)

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
1.7	0.95542 (-1.45e-05)	0.95647 (0.001035)	0.95742 (0.001985)	0.95841 (0.002975)	0.9593 (0.003865)	"0.960205 (0.004770)	0.96108 (0.005645)	0.96186 (0.006425)	0.96256 (0.007125)	0.963295 (0.007860)
1.8	0.96358 (-0.00049)	0.96437 (0.000300)	0.965235 (0.001165)	0.965915 (0.001845)	0.96673 (0.002660)	0.96751 (0.003440)	0.968255 (0.004185)	0.96901 (0.004940)	0.969705 (0.005635)	0.970385 (0.006315)
1.9	0.97167 (0.000387)	0.97223 (0.000947)	0.972915 (0.001632)	0.97354 (0.002257)	0.974085 (0.002802)	0.974665 (0.003382)	0.975365 (0.004082)	0.97596 (0.004677)	0.97656 (0.005277)	0.977155 (0.00587)
2.0	0.977155 (-9.49e-05)	0.977745 (4.95e-04)	0.978355 (1.11e-03)	0.978915 (1.67e-03)	0.97939 (2.14e-03)	0.979885 (2.64e-03)	0.980465 (0.003215)	0.980895 (0.003645)	0.98141 (0.004160)	0.981805 (0.004555)
2.1	0.982115 (-2.06e-05)	0.982625 (4.89e-04)	0.983015 (0.000879)	0.983445 (1.31e-03)	0.983825 (1.69e-03)	0.984265 (2.13e-03)	0.98464 (2.50e-03)	0.98497 (2.83e-03)	0.98534 (3.20e-03)	0.985725 (3.59e-03)
2.2	0.98602 (-7.66e-05)	0.986375 (2.78e-04)	0.98668 (0.000583)	0.98703 (9.33e-04)	0.987425 (1.33e-03)	0.987785 (1.69e-03)	0.988095 (2.00e-03)	0.988435 (2.34e-03)	0.98876 (2.66e-03)	0.989095 (0.002998)
2.3	0.98921 (-6.59e-05)	0.98954 (2.64e-04)	0.98979 (5.14e-04)	0.99 (7.24e-04)	0.990255 (0.000979)	0.990585 (1.31e-03)	0.99079 (1.51e-03)	0.99104 (0.001764)	0.991265 (1.99e-03)	0.991495 (2.22e-03)
2.4	0.99188 (7.75e-05)	0.992035 (2.33e-04)	0.992165 (3.63e-04)	0.99236 (5.58e-04)	0.99253 (7.28e-04)	0.99276 (9.58e-04)	0.99295 (1.15e-03)	0.99314 (1.34e-03)	0.99332 (1.52e-03)	0.993485 (1.68e-03)
2.5	0.993835 (4.47e-05)	0.99403 (2.40e-04)	0.994215 (4.25e-04)	0.99441 (6.20e-04)	0.99456 (7.70e-04)	0.99475 (9.60e-04)	0.994935 (1.14e-03)	0.995045 (0.001255)	0.995245 (1.45e-03)	0.995405 (0.001615)
2.6	0.99536 (2.12e-05)	0.99547 (1.31e-04)	0.995605 (2.66e-04)	0.99573 (3.91e-04)	0.99584 (5.01e-04)	0.99599 (6.51e-04)	0.99609 (7.51e-04)	0.996235 (8.96e-04)	0.99635 (1.01e-03)	0.996435 (1.10e-03)
2.7	0.99658 (4.70e-05)	0.99665 (1.17e-04)	0.99675 (2.17e-04)	0.99691 (3.77e-04)	0.99702 (4.87e-04)	0.99711 (5.77e-04)	0.997185 (6.52e-04)	0.99726 (7.27e-04)	0.99736 (8.27e-04)	0.99743 (8.97e-04)
2.8	0.99743 (-1.49e-05)	0.997545 (1.00e-04)	0.997625 (1.80e-04)	0.9977 (2.55e-04)	0.997755 (3.10e-04)	0.997785 (3.40e-04)	0.99789 (4.45e-04)	0.997985 (5.40e-04)	0.998035 (5.90e-04)	0.998105 (6.60e-04)
2.9	0.998035 (-9.92e-05)	0.998075 (-5.92e-05)	0.998155 (2.08e-05)	0.9982 (6.58e-05)	0.998265 (1.31e-04)	0.998305 (1.71e-04)	0.99836 (2.26e-04)	0.9984 (2.66e-04)	0.998445 (3.11e-04)	0.998495 (3.61e-04)
3.0	0.998545 (-0.000105)	0.99859 (-6.01e-05)	0.998645 (-5.10e-06)	0.99869 (3.99e-05)	0.998725 (7.49e-05)	0.99879 (1.40e-04)	0.998815 (1.65e-04)	0.998845 (1.95e-04)	0.998905 (2.55e-04)	0.99892 (2.70e-04)
3.1	0.999055 (2.26e-05)	0.99908 (4.76e-05)	0.999115 (8.26e-05)	0.999155 (1.23e-04)	0.99919 (1.58e-04)	0.999205 (1.73e-04)	0.999235 (2.03e-04)	0.999265 (2.33e-04)	0.999295 (2.63e-04)	0.999315 (2.83e-04)
3.2	0.999365 (5.21e-05)	0.999395 (8.21e-05)	0.999435 (1.22e-04)	0.999445 (1.32e-04)	0.999475 (1.62e-04)	0.99949 (1.77e-04)	0.999505 (1.92e-04)	0.999515 (2.02e-04)	0.999555 (2.42e-04)	0.99956 (2.47e-04)

Tabela 5 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Transformação Polar de Marsaglia. Nos parênteses, o valor encontrado via Transformação Polar menos o valor obtido pela pnorm() do R para o respectivo quantil (continuação)

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
3.3	0.99954 (2.34e-05)	0.99955 (3.34e-05)	0.99956 (4.34e-05)	0.99957 (5.34e-05)	0.99959 (7.34e-05)	0.999595 (7.84e-05)	0.999605 (8.84e-05)	0.999615 (9.84e-05)	0.999625 (1.08e-04)	0.99963 (1.13e-04)
3.4	0.999615 (-4.81e-05)	0.99963 (-3.31e-05)	0.999635 (-2.81e-05)	0.999655 (-8.07e-06)	0.99966 (-3.07e-06)	0.999685 (2.19e-05)	0.999685 (2.19e-05)	0.99969 (2.69e-05)	0.99971 (4.69e-05)	0.99972 (5.69e-05)
3.5	0.999745 (-2.24e-05)	0.99975 (-1.74e-05)	0.999755 (-1.24e-05)	0.99977 (2.63e-06)	0.99977 (2.63e-06)	0.999775 (7.63e-06)	0.99978 (1.26e-05)	0.99978 (1.26e-05)	0.99979 (2.26e-05)	0.999795 (2.76e-05)
3.6	0.99988 (3.91e-05)	0.999895 (5.41e-05)	0.999915 (7.41e-05)	0.999915 (7.41e-05)	0.999915 (7.41e-05)	0.999915 (7.41e-05)	0.99993 (8.91e-05)	0.99993 (8.91e-05)	0.999935 (9.41e-05)	0.999935 (9.41e-05)
3.7	0.999895 (2.80e-06)	0.999895 (2.80e-06)	0.999895 (2.80e-06)	0.9999 (7.80e-06)	0.999915 (2.28e-05)	0.999915 (2.28e-05)	0.999915 (2.28e-05)	0.99992 (2.78e-05)	0.999925 (3.28e-05)	0.999925 (3.28e-05)
3.8	0.99995 (2.23e-05)	0.999955 (2.73e-05)	0.999955 (2.73e-05)	0.999955 (2.73e-05)	0.99996 (3.23e-05)	0.99996 (3.23e-05)	0.99996 (3.23e-05)	0.99996 (3.23e-05)	0.999965 (3.73e-05)	0.999965 (3.73e-05)
3.9	0.999965 (1.31e-05)	0.999965 (1.31e-05)	0.999965 (1.31e-05)	0.99997 (1.81e-05)	0.99997 (1.81e-05)	0.99997 (1.81e-05)	0.999975 (2.31e-05)	0.999975 (2.31e-05)	0.999975 (2.31e-05)	0.999975 (2.31e-05)

Tabela 6 – Densidade acumulada da Normal(0,1) em $(0, b)$ obtida via Transformação Polar de Marsaglia. Nos parênteses, o valor encontrado via Transformação Polar menos o valor obtido pela pnorm() do R para o respectivo quantil (continuação)

7 Conclusões

7.1 Sobre o Gerador Congruencial para a Uniforme (0,1)

- A significância dos testes foi alta, o que significa que o gerador é eficaz em gerar uma amostra da U(0,1).

- Comentários sobre a baixa eficiência de implementação de quase todo algoritmo em R sempre serão oportunos, já que o mesmo tem foco maior na legibilidade do código e proximidade com a "linguagem natural"do que com velocidade. Como não foi pedido este tipo de análise do comando do trabalho, decidimos por não estender mais um ainda um trabalho que já ficou longo.

7.2 Sobre o Gerador usando a Transformação Polar de Marsaglia para a Normal (0,1)

- Importante notar que, como este Gerador utiliza o Gerador Congruencial para U(0,1) implementado aqui, era essencial que os Testes para ambos fossem significativos. E foram.

- Foram implementadas duas versões do gerador: uma que utiliza as funções **sin()** e **cos()** do próprio R e outra que utiliza a Transformação de Marsaglia. Note que esta segunda é necessariamente mais eficiente que a primeira, já que o cálculo de funções trigonométricas tem um custo não-negligenciável[5].

7.3 Sobre a tabela de Densidades Acumuladas da Normal(0,1) utilizando Método de Integração de Monte Carlo

- Sob uma ótica qualitativa, entendemos que os resultados foram satisfatórios. Temos uma maioria de diferenças que têm ordem 10^{-6}, com algumas excentricidades nos quantis a partir de 3, como as densidades acumuladas acima de 1 em 3.1, 3.6 e 3.7. Como estão acima de 1 mas espera-se de qualquer maneira que sejam muito próximas de 1, entendemos que isso não desabona o método.

- Uma observação importante sobre a construção da tabela: para fins estéticos, foi utilizada a função **format()** do R para especificar o número de dígitos do resultado, o que resulta em aproximações.

[5] Para uma discussão mais substancial sobre o tema, ver http://stackoverflow.com/q/2284860/4965975

Para o percentil da acumulada (o valor da célula sem os parênteses), foi utilizado o parâmetro **digits = 6**, ou seja, o 7o dígito foi arredondado. Para os erros (valores entre parênteses na 2a linha da célula), o parâmetro digits teve valor 3, pois o entendimento foi que a ordem de grandeza dos erros era mais relevante que o seu valor em si.

- O R utiliza uma implementação do pacote QUADPACK(**??**) em C, que por sua vez é uma implementação da Quadratura da Fórmula da Quadratura de Gauss-Kronrod para Integração Numérica, inclusive para **pnorm()** e similares. A impressão de que a Integração via MC é mais simples que este método é muito clara.

7.4 Sobre a tabela de Densidades Acumuladas da Normal(0,1) utilizando Transformação Polar de Marsaglia

- A implementação da Densidade Acumulada Empírica que foi utilizada para aproximar a Densidade Acumulada é, grosso modo, uma aplicação do Método de Rejeição. Isso resultou numa implementação fácil, mas diminui a robustez do procedimento quando o número de rejeições é muito baixo e há arredondamento.
- Em tempo, a construção da tabela foi feita de maneira muito similar à utilizada na tabela usando o Método de Monte Carlo.
- Ao contrário do Método de Monte Carlo, o Método Polar não resultou em percentis > 1, justamente pelo fato de ser uma implementação do Método de Rejeição.

Referências

RAMÍREZ, M. C. V.; ORSINI, J. A. M. *Desempenho dos modelos climáticos do IPCC em simular a precipitação presente e futura sobre o território brasileiro*. [S.l.]: Instituto Nacional de Pesquisas Espaciais–INPE, 2007. Citado na página 2.

WIKIPEDIA. *Pseudorandomness — Wikipedia, The Free Encyclopedia*. 2015. [Online; accessed 17-April-2016]. Disponível em: <https://en.wikipedia.org/w/index-.php?title=Pseudorandomnessoldid=696096676>. Citado na página 3.

8 Anexo A - Códigos das Implementações em R

Lembrete: repositório completo do Trabalho Prático, inclusive escrita do relatório, em https://github.com/august-o/TP2-EstComp

8.1 runif_congruencial.R

Implementa o Gerador Congruencial da U(0,1). O objeto .Random.seed é um vetor aleatório gerado pelo R a cada execução do interpretador.

```
runif_congruencial <- function(n = 1) {
  a <- 62089911
  m <- (2 ^ 31) - 1
  x <- c(0)
  y <- c(0)
  x[1] <- 0
  y[1] <- sample(.Random.seed[.Random.seed > 0], 1)
  if (n == 1) {
    y_0 <- y[1]
    y[1] <- (a * y_0) %% m
    x[1] = y[1] / m
  }
  else{
    y_0 <- y[1]
    y[1] <- (a * y_0) %% m
    x[1] = y[1] / m
    for (i in 2:n) {
      y[i] <- (a * y[i - 1]) %% m
      x[i] <- y[i] / m
    }
  }
  return(x)
}
```

8.2 rnorm_polar.R

Implementa o gerador usando o Método Polar para a N(0,1). **rnorm_polarTrig()** utiliza **sin()** e **cos()**, enquanto **rnorm_polarDireto()** utiliza as Transformações de Box-Muller/Marsaglia para um procedimento mais eficiente. Para fins de construção da tabela, foi utilizada a opção mais eficiente.

```
rnorm_polarTrig <- function(n = 1) {
```

```
2     resultado <- c()
3     i <- 1
4     while (i <= n) {
5       u1 <- runif_congruencial()
6       r2 <- -2 * log(u1)
7       u2 <- runif_congruencial()
8       theta <- 2 * pi * u2
9       x <- sqrt(r2) * cos(theta)
10      y <- sqrt(r2) * sin(theta)
11      resultado <- append(resultado, c(x, y))
12      i <- i + 1
13    }
14    return(resultado)
15  }
16
17  rnorm_polarDireto <- function(n=1){
18    X <- matrix(0, ncol = 2, nrow = n)
19    for (i in 1:n){
20      U <- runif(2)
21      V <- 2 * U - 1
22      R2 <- sum(V^2)
23      while (R2 > 1){
24        U <- runif(2)
25        V <- 2 * U - 1
26        R2 <- sum(V^2)
27      }
28      Y <- sqrt(-2 * log(R2) / R2)
29      X[i, ] <- Y * V
30    }
31    as.vector(X)
32  }
```

8.3 pnorm_MC.R

Implementa o Método de Integração de Monte Carlo para obter uma aproximação para a Densidade Acumulada da N(0,1) na forma $P(0 < Z < b)$.

NOTA: Como este código foi utilizado para redigir o trabalho, o retorno da função é uma lista com o valor encontrado para fda usando MC e caracteres especiais do Latex que auxiliaram na construção da tabela, seguido pela diferença entre o valor via MC e via pnorm() do R.

```
pnorm_MC <- function(x, n = 10^6){
  t <- runif(n)
  fda <- numeric(length(x))
  resultado <- list(length(x))
  for(i in 1:length(x)){
    g <- x[i]*exp(-0.5*(x[i]*t)^2)
    fda[i] <- mean(g)/sqrt(2*pi) + 0.5
    resultado[i] <- paste(format(fda[i],digits = 6),"
␣␣␣␣\newline(",format(fda[i]-pnorm(x),digits=3),")&",sep="")
  }
  return(resultado)
}
```

8.4 pnorm_polar.R

Implementa uma aproximação para a Densidade Acumulada da N(0,1) utilizando Transformação Polar de Marsaglia e Método da Rejeição. Ver **NOTA** acima para explicação sobre o retorno da função.

```
        pnorm_polar <- function(x,n=10^5){
  resultado <- list(length(x))
  fda <- numeric(length(x))
  t <- rnorm_polarDireto(n)
  mean(ifelse(t<x,1,0))
  for(i in 1:length(x)){
    fda[i] <- mean(ifelse(t<x[i],1,0))
    resultado[i] <- paste(format(fda[i],digits = 6),
    "\newline(",format(fda[i]-pnorm(x),digits=3),")&",sep="")
  }
  return(resultado)
```